

CMPE 12/L Syllabus - Spring 2018

Computer Systems and Assembly Language

Prof. Matthew Guthaus
University of California, Santa Cruz

April 2, 2018

Overview

Introduction to computer systems and assembly language and how computers compute in hardware and software. Topics include digital logic, number systems, data structures, compiling/assembly process, basics of system software, and computer architecture. Prerequisite(s): previous or concurrent enrollment in course 12L is required.

We will be using the MIPS assembly language throughout the class and using MARS as a simulator. **This class has a prerequisite of prior programming experience (such as CMPS 5C/J/P).** While we will be writing code in MIPS assembly, knowledge of programming is required to succeed.

The class is fast paced due to the breadth of material covered; keep up with the class and do the readings. Do NOT wait to start your programming assignments until the day before they are due; most students who have difficulty in this class wait until the last minute and don't have sufficient time to get help.

Instructor

- Prof. Matthew R. Guthaus
 - Email: mrg+ce12@ucsc.edu (all questions should be asked on Piazza)
 - Office Hours: Tuesday 2-3pm in E2-225 or by appointment

Class Times

- Lecture: TuTh 11:40am - 1:15pm, Media Theater M-110
- Lab Sections (one only):
 - 01A MW 09:00AM-10:00AM Loc: MingOng Cmp Lb 103
 - 01B MW 08:00PM-09:00PM Loc: MingOng Cmp Lb 103
 - 01C TuTh 09:00AM-10:00AM Loc: MingOng Cmp Lb 103
 - 01D TuTh 10:00AM-11:00AM Loc: MingOng Cmp Lb 103
 - 01E TuTh 02:00PM-03:00PM Loc: MingOng Cmp Lb 103
 - 01F TuTh 03:00PM-04:00PM Loc: MingOng Cmp Lb 103
 - 01G TuTh 06:00PM-07:00PM Loc: MingOng Cmp Lb 103
 - 01H TuTh 07:00PM-08:00PM Loc: MingOng Cmp Lb 103
 - 01I TuTh 08:00PM-09:00PM Loc: MingOng Cmp Lb 103
 - 01J WF 10:00AM-11:00AM Loc: MingOng Cmp Lb 103

Teaching Assistants

- Megan Boivin (mcboivin@ucsc.edu)

- Carlos Espinosa (caiespin@ucsc.edu)
- Cory Ibanez (cibanez@ucsc.edu)
- Michael Powell (mopowell@ucsc.edu)
- Rebecca Rashkin (rrashkin@ucsc.edu)

Online Resources

The main web site for this course will be at:

<https://canvas.ucsc.edu>

This includes lecture slides, assignments and grades. In order to access this, you must have a valid UCSC login ID. Additional announcements will be made at the beginning of lecture or sent to your UCSC email address.

Required text books

The first is an abridged version of the textbook previously used for a much cheaper price (if you have access to the full book the chapters taken are identical). The second is a much more detailed book on Computer Architecture, but may be useful in this and later courses. The last is a free MIPS programming book.

- Yale N. Patt and Sanjay J. Patel, Introduction to Computing Systems: From Bits and Gates to C and Beyond (Reader), McGraw-Hill Education, First Edition, ISBN 978-1307117530, 2017.
- David A. Patterson and John L. Hennessy, Computer Organization and Design MIPS Edition: The Hardware/Software Interface, Fifth Edition, ISBN 978-0124077263, 2013.
- Charles W. Kann, Introduction to MIPS Assembly Language Programming, Gettysburg College Open Educational Resources, 2015. <http://cupola.gettysburg.edu/oer/2/>

Key Dates

- Midterm Exam: Tuesday, May 15 (during class)
- Final Exam: Monday, June 11, 8am (in class)

Class Evaluation

The class will be evaluated using the following weights of grades:

- Labs 50%
- Midterm 25%
- Final 25%

You will receive the same grade for both the lecture and the lab. The material covered in the lab is meant to reinforce the material presented in class. Thus doing well in the lab will directly improve your performance on exams and homework. Assume that CMPE 12 and 12L are one 7-unit course. If you fail the lecture, you will also receive a failing grade in lab.

You must take both the lab and lecture to complete this course. The material in the lecture and lab is tightly coupled and it is not possible to take one without the other. All students must complete all of the labs and take both exams.

It is not possible to pass this course with a 0 on any lab assignment. You can receive a zero if the lab assignment was not submitted, files are missing, you do not do the lab, or you are caught cheating.

In order to pass the class, you will need to demonstrate basic mastery of the subject matter as demonstrated by your labs and exams; no one may pass the class with less than 50% average of your midterm and final exam scores. Likewise your average lab scores must be above 50% to pass the class. Note that these are not sufficient to pass the class, but you cannot pass the class without meeting these minimal criteria (necessary but not sufficient).

Re-grading of lab assignments will only be done if we have a clerical error (i.e., we added points wrong) or we somehow missed your work. Note that this does not include you forgetting to submit or submitting the wrong commit ID in the Google Form for your lab.

Lab Requirements

The labs are to a large extent where you will master the class subject matter. Note that due to the size of the class, most of the assignments will be machine graded. As such, you will be required to comply with the auto-grading software in order to get any points in the lab. There is a hard requirement on each and every lab to have the correct files in the correct directories.

Each lab will specify a set of MINIMUM SUBMISSION REQUIREMENTS that will be checked automatically. You will receive an email after the due date and will be able to use your grace period to fix any issues. Failure to meet this requirement will result in failing the class regardless of any other scores. Typically this requirement will be to have the correctly named files in the correctly named directories but you should check each lab assignment carefully to make sure you have precisely (not approximately) followed its MINIMUM SUBMISSION REQUIREMENT.

We have set up a webpage that will check your submission for the MINIMUM SUBMISSION REQUIREMENTS that you can use to verify that you have, in fact, turned in the correct files. See the CMPE012_GIT.pdf document for details.

Again, having these files in the correct place and with the correct names does not mean you pass the lab, but is a necessary requirement (necessary but not sufficient). These files need to be there for each and every lab to pass the class. Because of the size of the class, there is no flexibility on this. If you cannot master these simple requirements, then you do not have the required prerequisite knowledge to succeed in this class.

All labs will require specific input and output formatting. Again, due to the nature of auto-grading, formats need to be adhered to. Each lab assignment will have the input/output specification called out, and will have at least three examples in the assignment. If there are ambiguities, ask on Piazza and they will be resolved.

Assignments are designed to take more time than your lab section to complete; the lab section is to give you guidance and help, it is expected that you will go off and program on your own both before and after sections. The total time given to complete the labs is sufficient if you start early; starting just before it is due is a recipe for failure.

Using GIT

We will be using GIT for version control and also for lab submissions; this is an industry standard tool for version control and learning it is an important part of the class. You cannot pass this class without submitting assignments using GIT.

We have a supplementary document on using GIT, which explains the mechanics of using it specifically in this class. It is also how we will be having you turn in your lab assignments for grading (both pushing it to the server and submitting the Commit ID on a Google form). Pushing the code to the server safeguards it from loss, and the Commit ID tells us which version of your code you would like us to grade and when you submitted it. Note that both are required to turn in your assignment.

You are encouraged to commit and push your repo early and often. There is no penalty for doing so, and it will make sure that you have met the minimum file requirements for the lab. Learn to use GIT wisely, check your code frequently (and start it early). A small portion of your lab grade may be based on how often you committed your code using GIT.

Likewise, you should always remember to submit your Commit ID for each assignment in that assignment's Google form. Your assignment is not submitted until this is done and you will have to use late hours to submit after the deadline.

Grace Period

We understand that sometimes things in life occur that are beyond your control, and you will need extra time not because you waited until the last minute to start. In order to mitigate this each student is granted 72 hours of “grace period” to be used throughout the quarter on your lab assignments. It is consumed in 12 hour (half day) increments, and you will be able to check the “late days” assignment on CANVAS to see how many hours you have left. Late time includes weekends, holidays, and any time taken after the due date.

Specifically, we do NOT need to know why you are using your grace period. If there is some extenuating circumstance that is not covered by the grace period, we need to be informed of this ASAP (not after the fact).

The time stamp of your Google form submission determines when your assignment is submitted. This form requires the GIT Commit ID to submit and your name. Your assignment is not submitted until you submit the Commit ID and you will use late days until you do submit it. You may resubmit a new Commit ID at any time by clicking the edit link in the confirmation email.

Getting Help

The primary place to get help in this course is your lab section. This section will have a TA and several tutors to assist students one-on-one with issues. Do not wait until the last minute, however, because there may be long lines to get help.

In addition to in-person help, questions may be asked in the Piazza forum at:

<https://piazza.com/ucsc/spring2018/cmpe121>

This is monitored by the TAs and the instructor. If you have any need to contact the teaching staff, it should be done through Piazza. You may select a “Private Post” and the message on Piazza will go to the Instructor and TAs only.

Do not post code publicly in Piazza. This will be considered a form of academic misconduct. If you are uncertain, you may share a selection of your code with the teaching staff using a Private Post.

Direct e-mail to the Professor should only be used for extreme emergencies. This is a large class, so expect a delay in response. Any such email should be sent to mrg+ce12@ucsc.edu or it will be immediately filtered and deleted. Any non-emergency emails will be directed to the Piazza forum. **Piazza is the required method of communication for this course.**

Note that Piazza is a public, professional forum; please treat your communication with others with a modicum of dignity and respect. You are not anonymous. You ARE training to be a professional, treat each other with respect.

Academic Integrity

You may consult with me, the TA/tutor and other students, verbally and possibly with the use of a board. However, you must be able to produce the solution to the problem without any notes from this consultation. You should spend some time thinking about a problem before consulting anyone – this is how to learn. You must cite any materials (besides the text and course handouts) that you use to solve the problems.

No collaboration is allowed on lab assignments. You are not allowed at any point to share actual code with another student. To ensure that you do not “accidentally” copy, you should not take any notes or exchange any written or electronic information from discussions that you have with another student. My guideline is: You can discuss a problem and not take notes, but wait 15 minutes before

working on your solution. This ensures that your code will be your own. **It is also cheating to allow your work to be copied.**

There is zero tolerance of academic dishonesty. In the event that we detect cheating, you will receive a zero for that assignment or lab or exam. For any case, you will be reported to your college Provost for a formal investigation.

Your Lab0 assignment will require you to check in this syllabus and the Personal Responsibility document into the repo and indicate that you have read it. No further work will be accepted from any student who has not completed this lab on time.

Incompletes

Incompletes will be given only for medical and family emergencies. In this event, you must be in good standing (i.e. have a passing grade so far) and request an incomplete from the instructor before the last day of instruction. NOTE: If you ask for an incomplete after not turning in labs or homework assignments, you are not in good standing.

Disability Accomodations

If you would like to request academic accommodations due to a disability, please contact the Disabled Resource Center, 146 Hahn Student Services, (831)459-2089(voice) (831)459-4806(TDD/TTY). They can authorize specific accommodations for you on an Accommodation Request Form. Please discuss this form with the instructor in the first 2 weeks of the quarter so that we can discuss the accommodations you need for class. You will need to confirm with SOE teaching services at least one week before each exam to secure testing accomodations.

Topics (Approximate)

- History and Introduction
- Numbering Systems
- Binary Numbers
- Data Representation
- Digital Logic
- Digital Logic Gates in Transistors
- Digital Logic Structures
- ALU Computations
- Computing Overview
- MIPS ISA
- Intro MIPS Programming
- Arrays and Stacks
- Function Calls / Macros
- MIPS Instruction Decoding
- MIPS Architecture
- IO and Exceptions / Traps / Interrupts
- Number systems revisited
- Fractional Binary
- Floating Point Representation
- Floating Point Arithmetic
- Sequential Logic / Boolean Algebra